Beijing Forest Studio
北京理工大学信息系统及安全对抗实验中心

预训练语言模型GPT3

硕士研究生 高依萌

2021年02月07日

内容提要

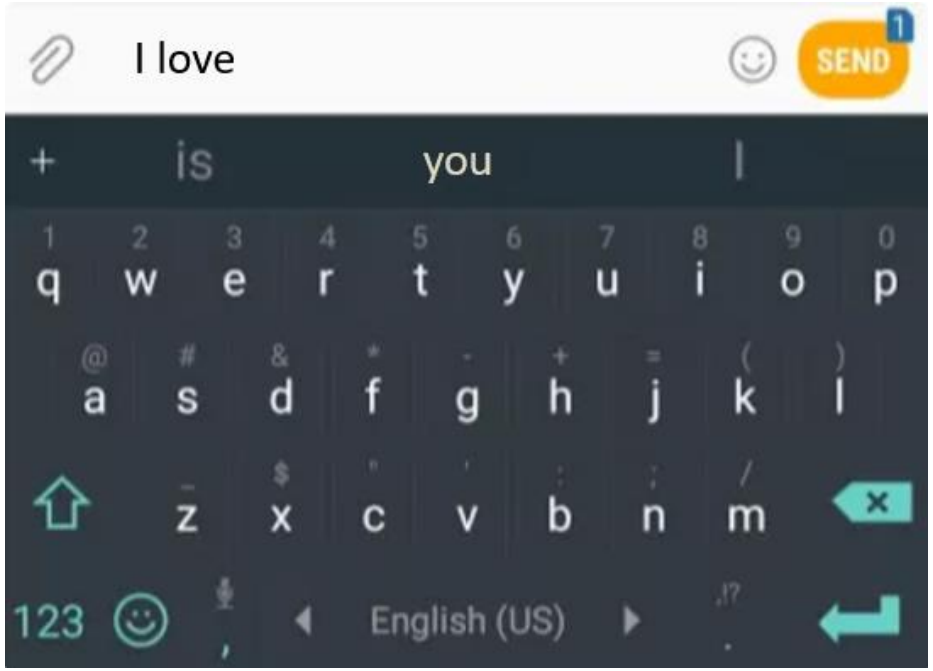
- 背景简介
- 基本概念
- 算法原理
- 优劣分析
- 应用总结
- 参考文献

背景简介

- 预期收获
  - 1. 理解词嵌入，预训练语言模型基本概念，attention算法，transformer基本结构
  - 2. 梳理GPT1到GPT3的模型结构变化
  - 3. 了解GPT3的优缺点
  - 4. 了解GPT3在NLP领域的应用

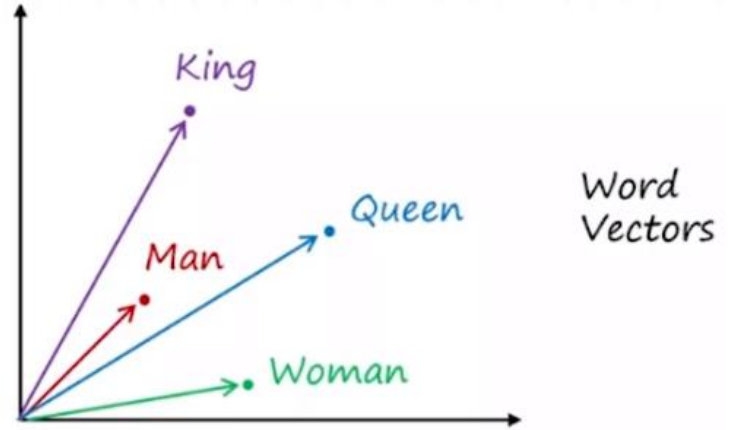
背景简介-问题引入

- 输入法
  - 如何根据输入内容，提示下一个单词?
    - 单词如何表示?
    - 预测单词的模型如何构建?



背景简介-词的表示

- 离散表示(one-hot representation)
  - 维数灾难：高维稀疏，不易训练
  - 语义鸿沟：没有刻画词之间的语义相似度
- 分布式表示，词嵌入(word embedding)
  - 低维稠密，易于训练
  - 欧式距离刻画语义相似度



分布式向量在二维空间映射示意图

$$\vec{King} - \vec{Man} + \vec{Woman} = \vec{Queen}$$

$$\begin{matrix} \text{king} \\ \text{queen} \\ \text{woman} \\ \text{man} \\ \vdots \end{matrix} \begin{bmatrix} 1\,0\,0\,0\,0\,0\ \ldots \\ 0\,1\,0\,0\,0\,0\ \ldots \\ 0\,0\,1\,0\,0\,0\ \ldots \\ 0\,0\,0\,1\,0\,0\ \ldots \\ \vdots \end{bmatrix}$$

One-hot

语言模型

训练

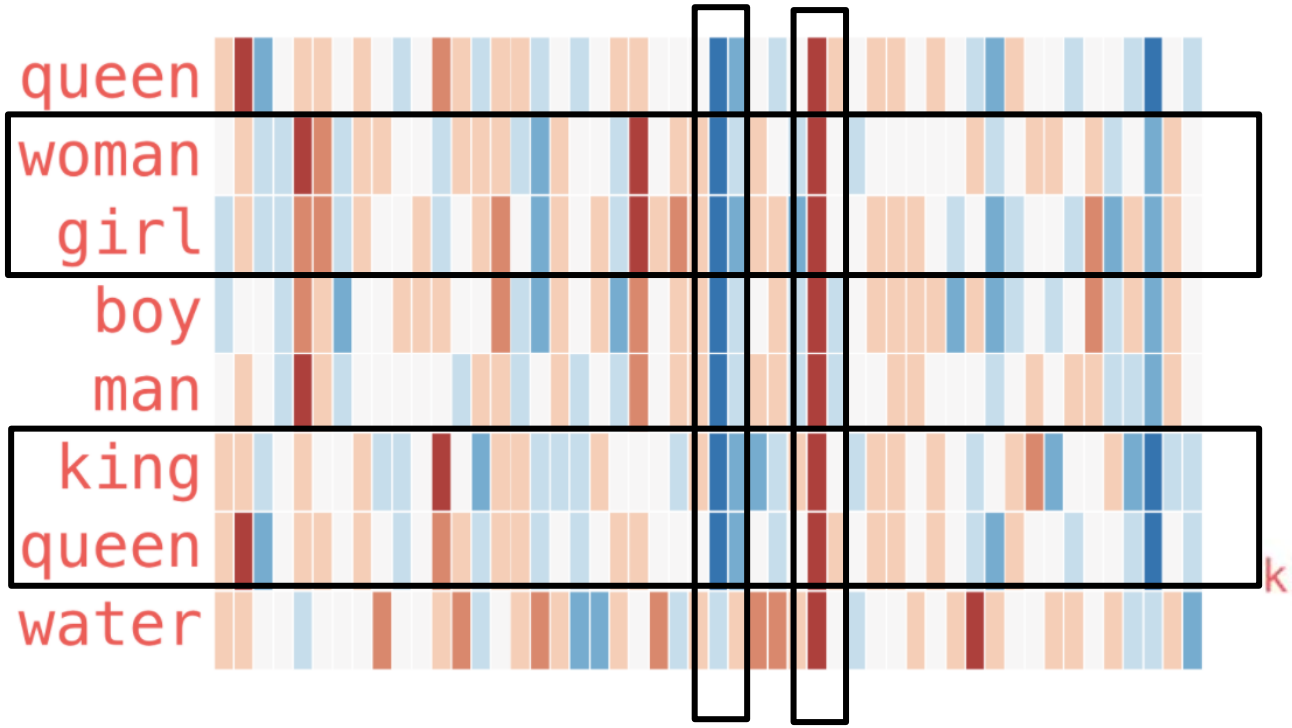
$$\begin{matrix} \text{king} \\ \text{queen} \\ \text{woman} \\ \text{man} \\ \vdots \end{matrix} \begin{bmatrix} 0.99\ 0.99\ 0.23\ -0.24\ \ldots \\ 0.99\ 0.56\ 0.23\ -0.32\ \ldots \\ 0.34\ -0.14\ 0.45\ 0.23\ \ldots \\ -0.33\ 0.55\ 0.23\ -0.35\ \ldots \\ \vdots \end{bmatrix}$$

distributed

词语向量化

背景简介-词嵌入可视化

- 数值转化成不同深浅的颜色
  - king [0.50451 , 0.68607 , −0.59517 , −0.022801, 0.60046 , −0.13498 , −0.08813 , 0.4 · · · · ·]



- 如果接近2，则为红色
- 如果接近0，则为白色
- 如果接近-2，则为蓝色

king − man + woman ~= queen

背景简介-统计语言模型

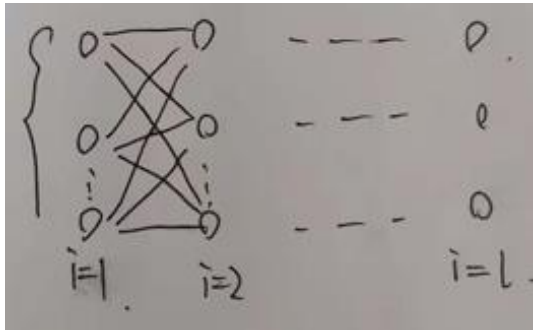
- 语言模型
  - 计算字符串w作为一个句子出现的概率
- 统计方法语言模型
  - $p(w)$
  - $= p(w_1, w_2, \cdots w_n)$
  - $= p(w_1)p(w_1|w_2)p(w_3|w_1w_2)\cdots p(w_l|w_1\cdots w_l)$
  - $= \prod_{i=1}^{l} p(w_i|w_1\cdots w_{i-1})$
  - $p(w_i|w_1\cdots w_{i-1}) \approx p\left(w_i\middle|w_{i-(n-1)},\cdots w_{i-1}\right)$
  - $p\left(w_i\middle|w_{i-(n-1)},\cdots w_{i-1}\right) = \frac{count(w_{i-(n-1)},\cdots,w_{i-1},w_i)}{count(w_{i-(n-1)},\cdots,w_{i-1})}$

词频相乘

w是一个句子，w_i是句子中的第i个单词

- 缺点
  - 参数过大
    - 很难查找最优路径
  - 零概率现象
    - 连乘

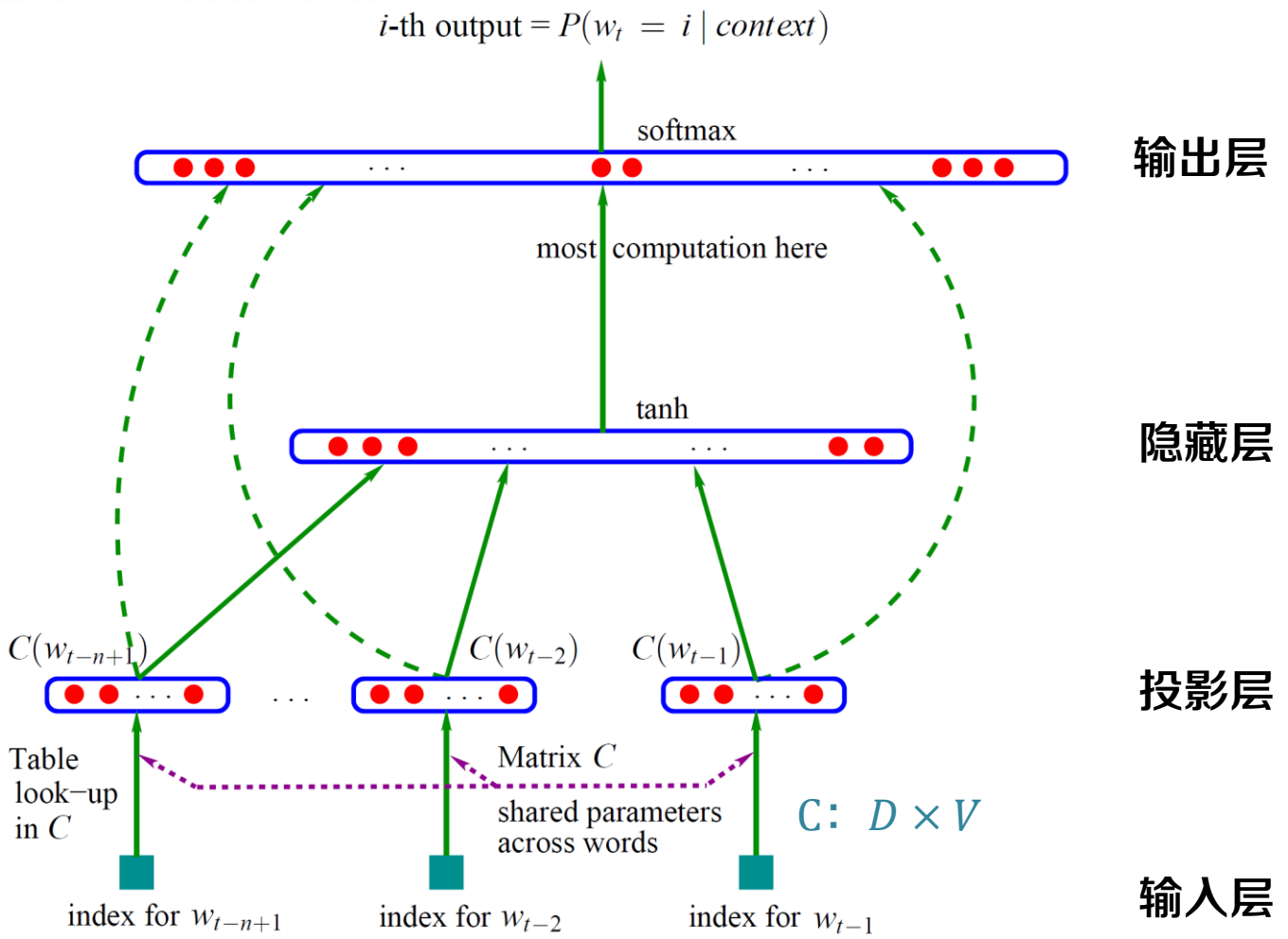
词汇表长度L



共L^i条路径

背景简介-神经网络语言模型(NNLM)

- NNLM
  - 参数共享
  - 单词预测融合前文信息
- 预训练语言模型
  - 已有模型参数对模型初始化
  - 模型二次训练微调参数



NNLM模型结构示意图

背景简介-历史发展

- 研究历史

百度IDL徐伟提出用神经网络训练语言模型

Mikolov提出Word2vec模型

OpenAI提出Gpt-1

OpenAI提出Gpt-2

Bengio提出经典神经网络语言模型(NNLM)

华盛顿大学提出ELMo

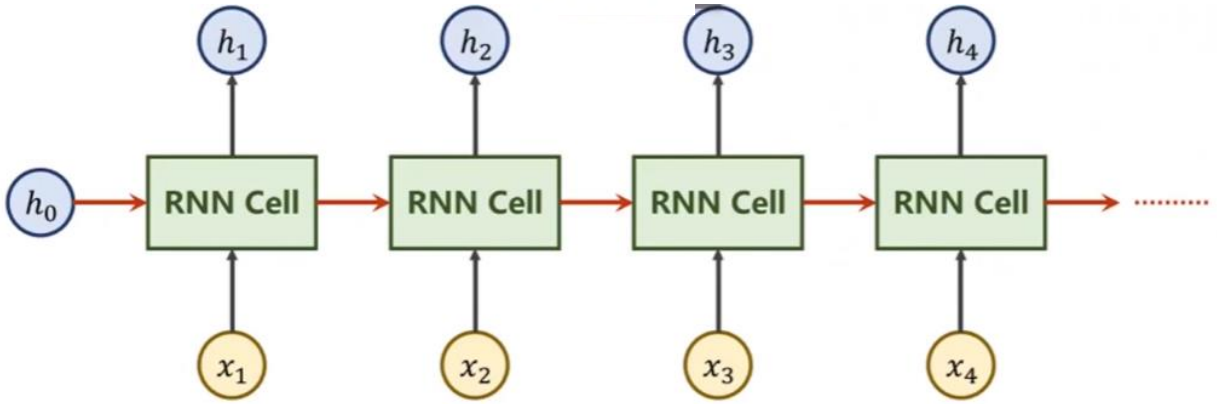
谷歌提出Bert模型

OpenAI提出Gpt-3

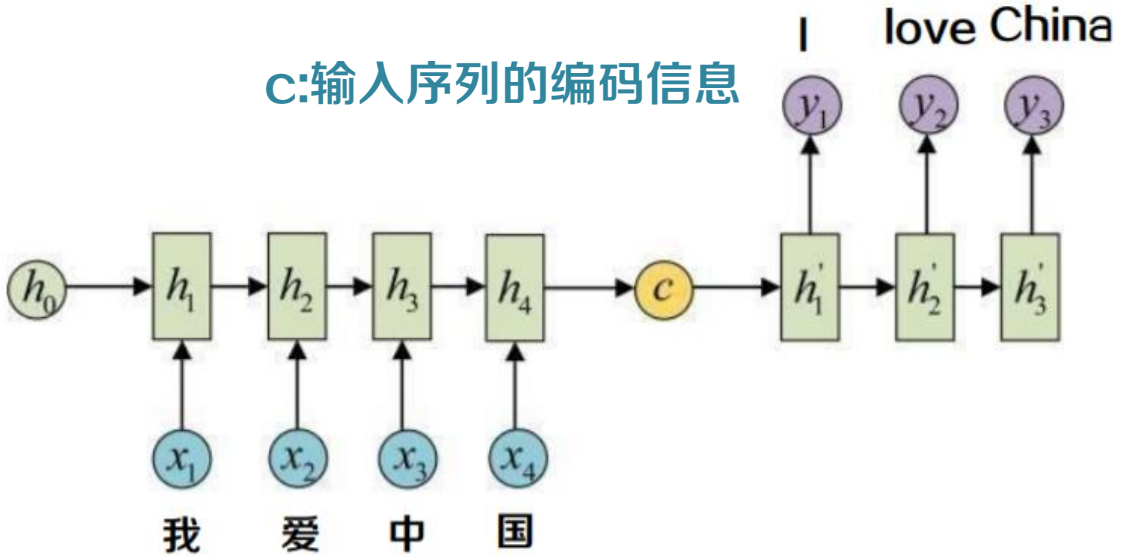
基本概念

基本概念-RNN

- RNN
  - 序列特性，无法并行处理输入数据
  - 梯度消失，无法获取长距离信息



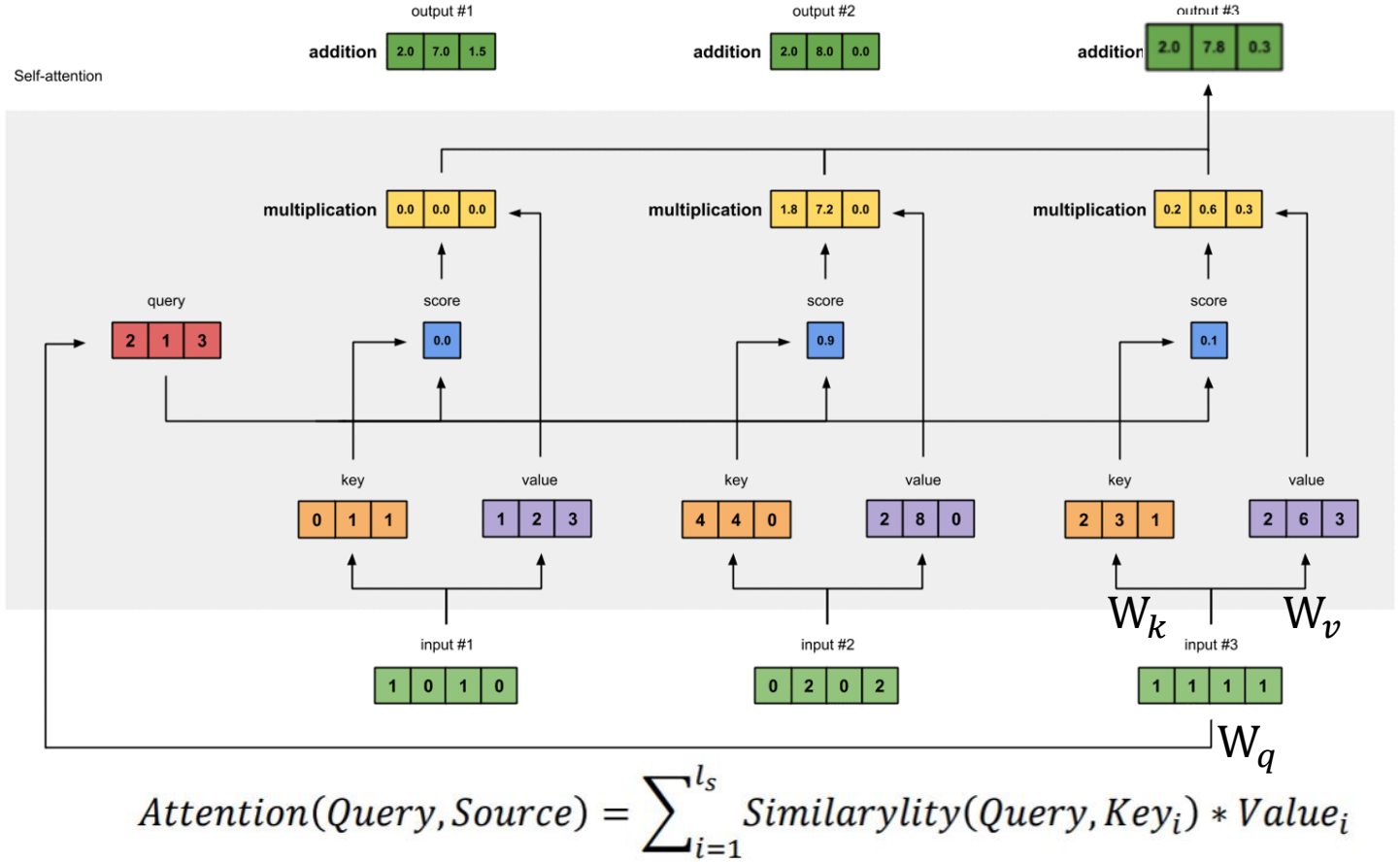
RNN结构示意图



翻译任务中RNN的应用

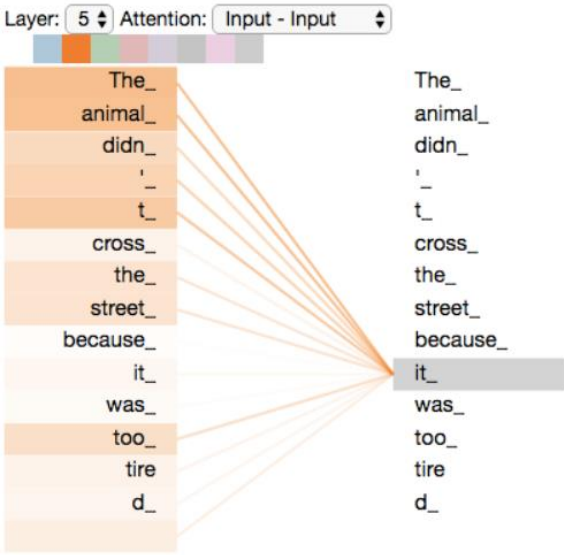
基本概念-注意力机制Attention

- 目的
  - 获取全局信息
  - 序列中的每个输出，预测输入标记对输出影响程度



$$Attention(Query, Source) = \sum_{i=1}^{l_s} Similarylity(Query, Key_i) * Value_i$$

"The animal didn't cross the street because it was too tired"

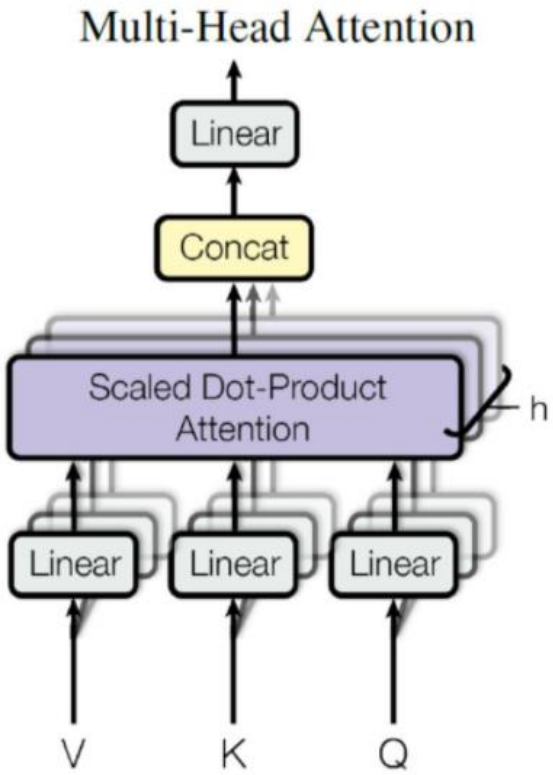


注意力可视化示意图

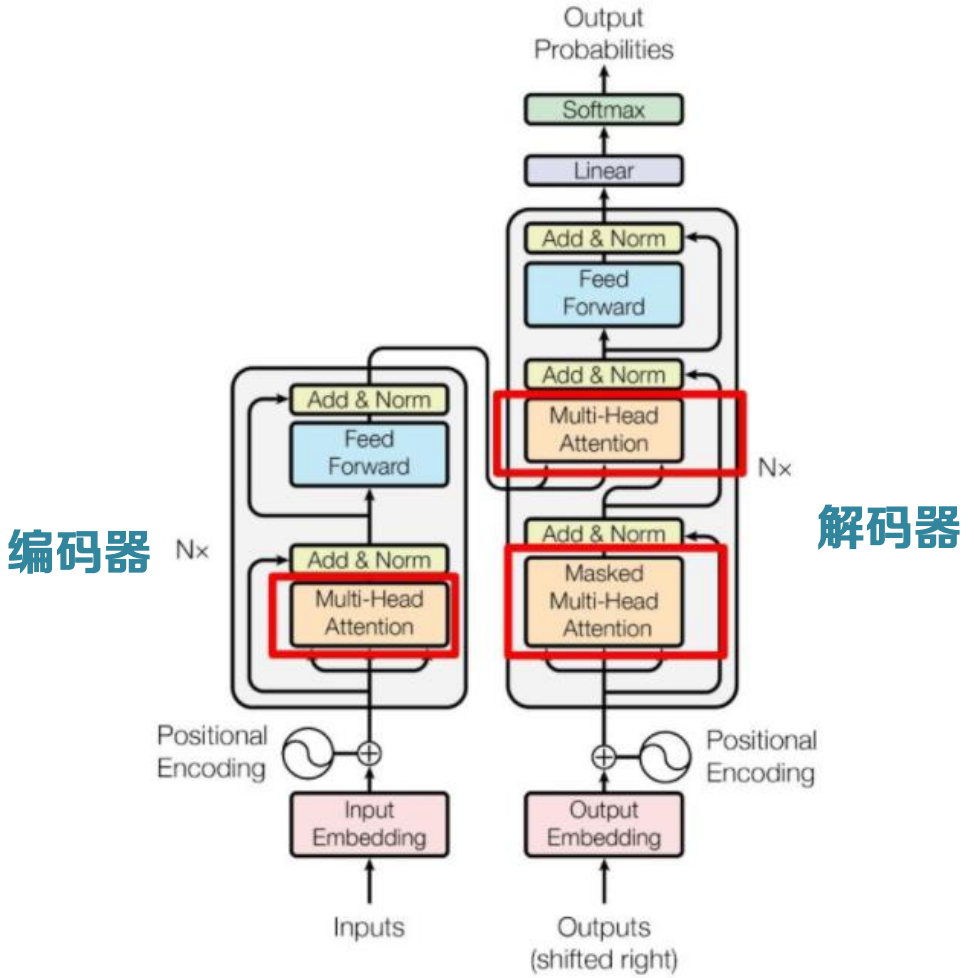
基本概念-transformer基本结构

- Multi-Head Attention

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



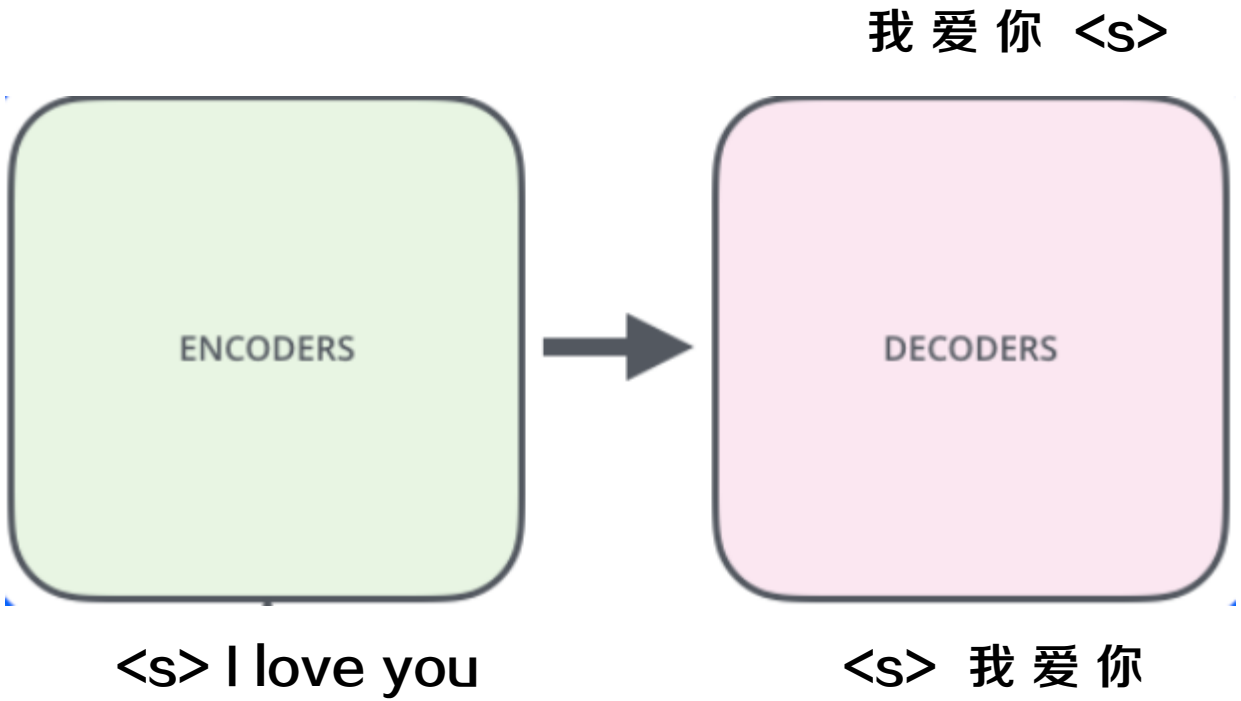
Seq2seq model

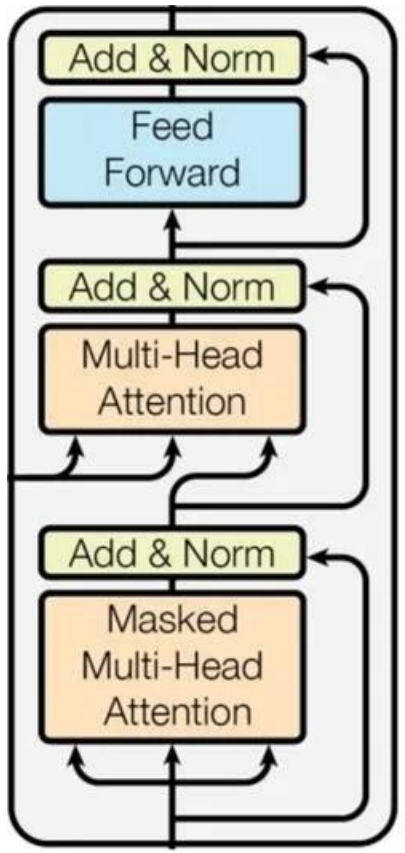


Transformer结构示意图

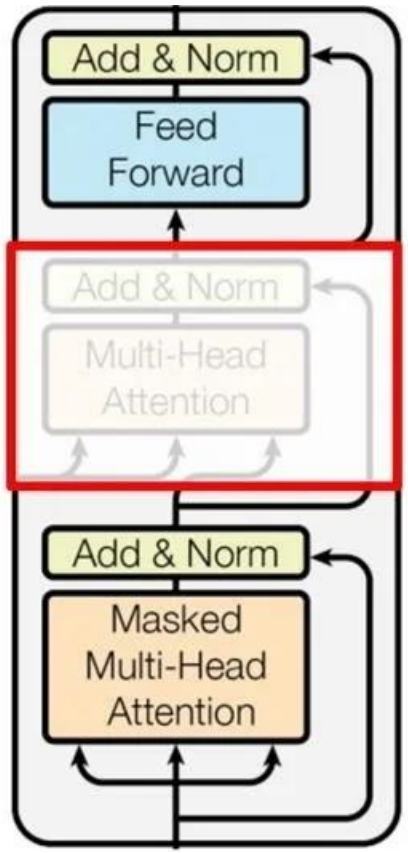
基本概念-GPT基本结构

- Masked操作的作用
  - 防止模型训练过程中获取预测结果





Transformer Decoder



GPT Decoder
去掉第二个 Multi-Head

基本概念-GPT基本结构

	A	B	C	D
A	0.11	0.00	0.81	0.79
B	0.19	0.50	0.30	0.48
C	0.53	0.98	0.95	0.14
D	0.81	0.86	0.38	0.90

Softmax 前

	A	B	C	D
A	0.11	-inf	-inf	-inf
B	0.19	0.50	-inf	-inf
C	0.53	0.98	0.95	-inf
D	0.81	0.86	0.38	0.90

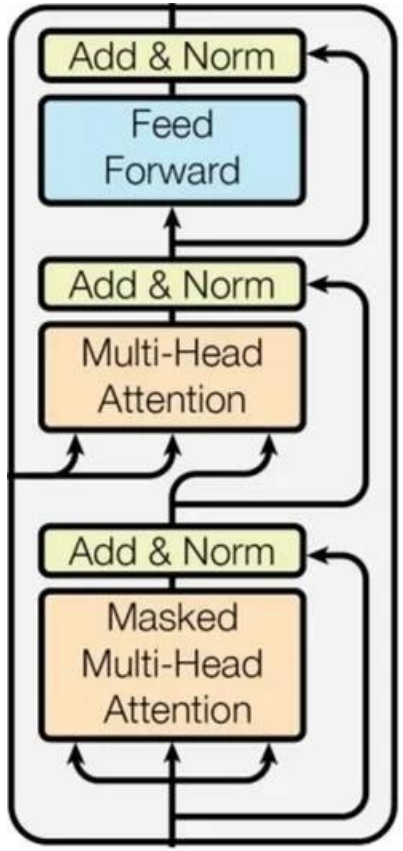
Masked

	A	B	C	D
A	0.11	-inf	-inf	-inf
B	0.19	0.50	-inf	-inf
C	0.53	0.98	0.95	-inf
D	0.81	0.86	0.38	0.90

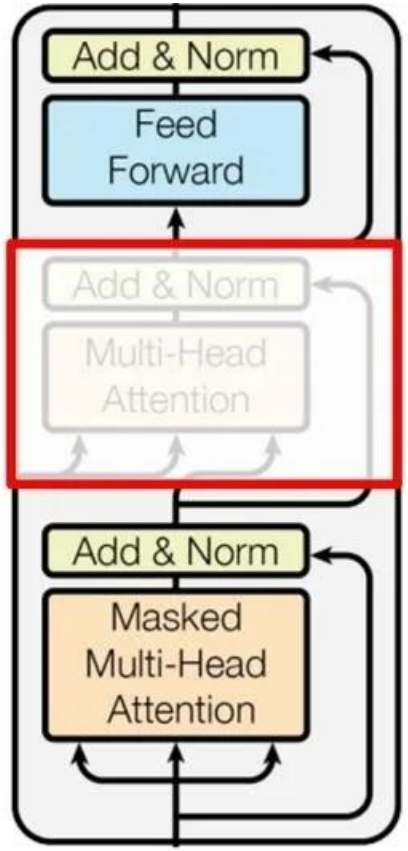
Masked

	A	B	C	D
A	1	0	0	0
B	0.48	0.52	0	0
C	0.31	0.35	0.34	0
D	0.25	0.26	0.23	0.26

Softmax后



Transformer Decoder



GPT Decoder

去掉第二层Multi-Head

算法原理

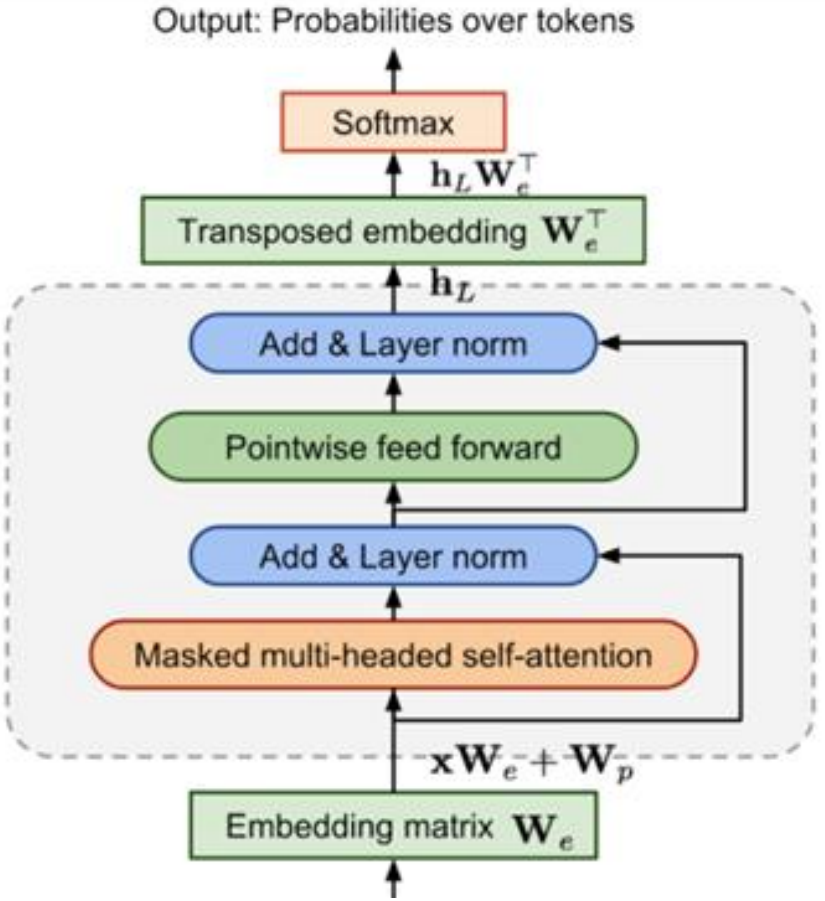
算法原理

T	通过语言模型训练获取词的分布式表示
I	输入前文单词的one-hot表示
P	堆叠多层transformer的decoder或其变形结构进行处理
O	预测下一个单词

P	如何通过上文预测下文单词
C	需要极其庞大训练集数据
D	训练成本极大
L	NeurIPS 2020

算法原理-GPT模型训练

- 模型结构
  - $h_0 = XW_e + W_p$
    - X是上下文词向量
    - W_e是词向量矩阵
    - W_p是位置向量矩阵
  - $h_l = transformer_{block(h_{l-1})} \forall i \in [1, n]$
    - 12层单向Transformer，n=12
  - $P(u) = softmax(h_n W_e^T)$
- 最大似然估计
  - $L_1(C) = \sum_i logP(x_i | x_{i-k}, \cdots, x_{i-1}; \theta)$
    - k是上文窗口



× 12

GPT-1模型结构

算法原理-Fine-tuning（微调）

- Supervised fine-tuning
  - 应用于有监督目标任务
  - 流程
    - 输入预训练好的模型
    - 获取最后输出向量h_l^m
    - 线性层和softmax预测标签

$$P(y|x_1, x_2, \cdots, x_m) = softmax(h_l^m W_y)$$

    - 损失函数

$$\mathrm{L}_2(C) = \sum_{(x,y)} logP(y|x_1, \cdots, x_m)$$

- 下游使用的监督模型目标函数

$$L_3(C) = L_2(C) + \lambda * L_1(C)\ \lambda = 0.5$$

辅助训练，缓解过拟合

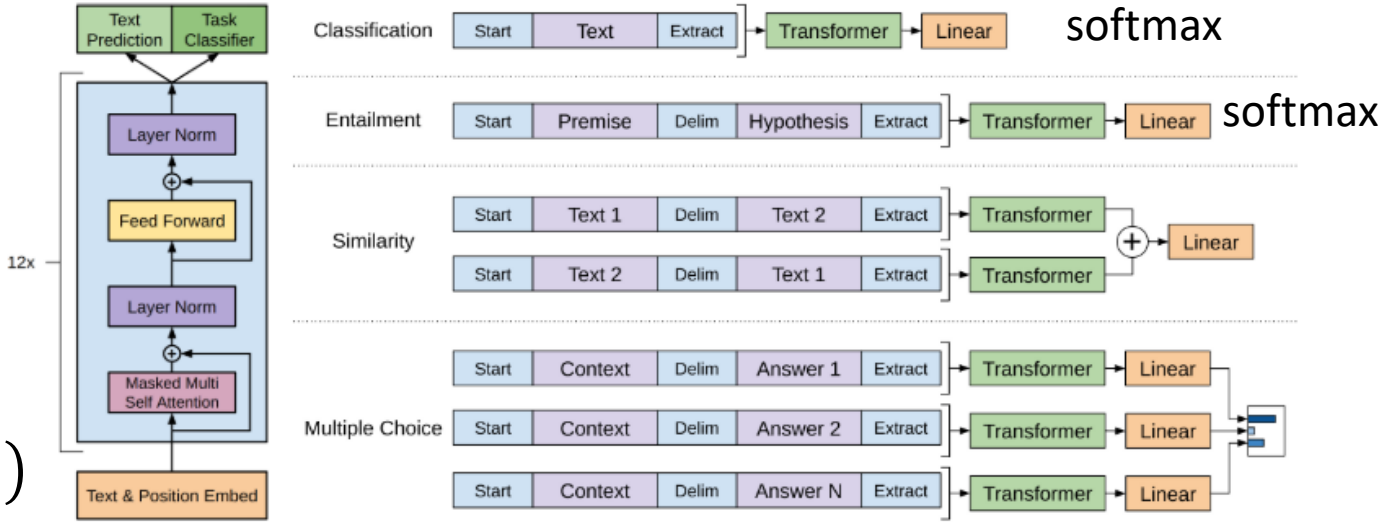
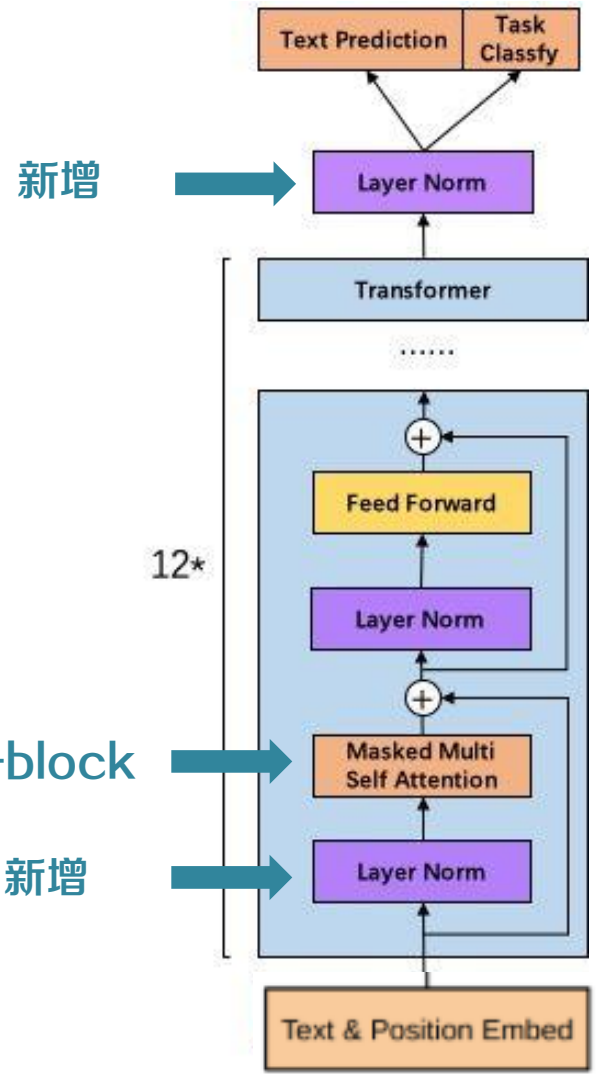


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

GPT-1下游任务结构示意图

算法原理-GPT-2

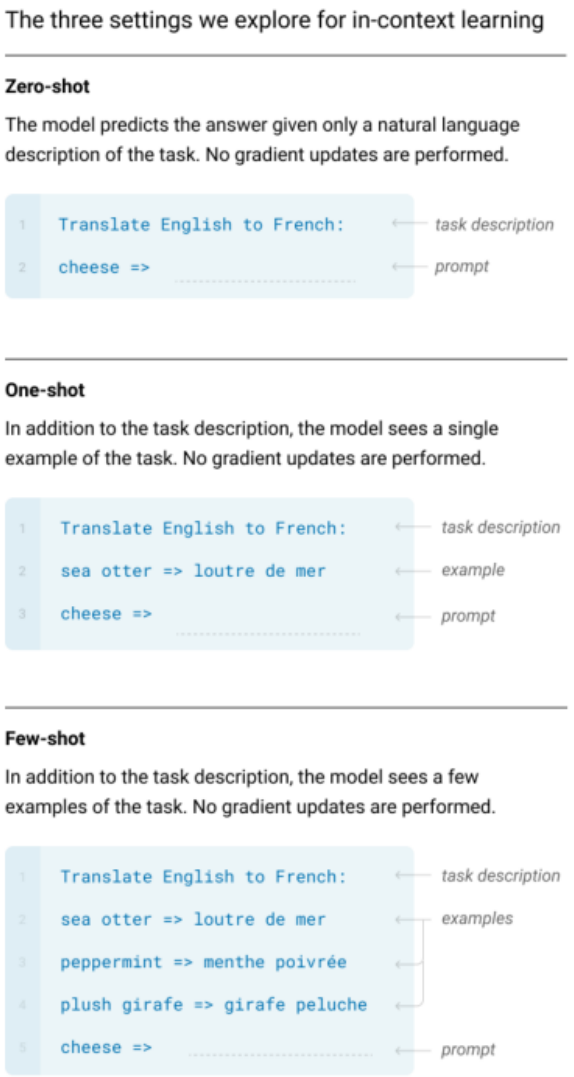
- 对比GPT-1改进
  - GPT-2去掉了fine-tuning
    - 模型自动识别需要做什么任务
  - 增加数据集规模
    - 训练集800万网页，40G
  - 增加网络参数
    - 48层transformer，隐层维度1600，15亿参数量
  - 调整transformer结构
    - 将layer normalization放到每个sub-block之前
    - 最后增加一个layer normalization



GPT-2基本结构图

算法原理—GPT-3

- GPT-3
  - 沿用GPT-2的模型框架
  - 采用不同学习方法
    - Few-shot
  - 增加网络参数
    - 96层transformer
    - 隐层维度12,288
    - 1750亿参数量
  - 稀疏注意力机制
    - 随机attention

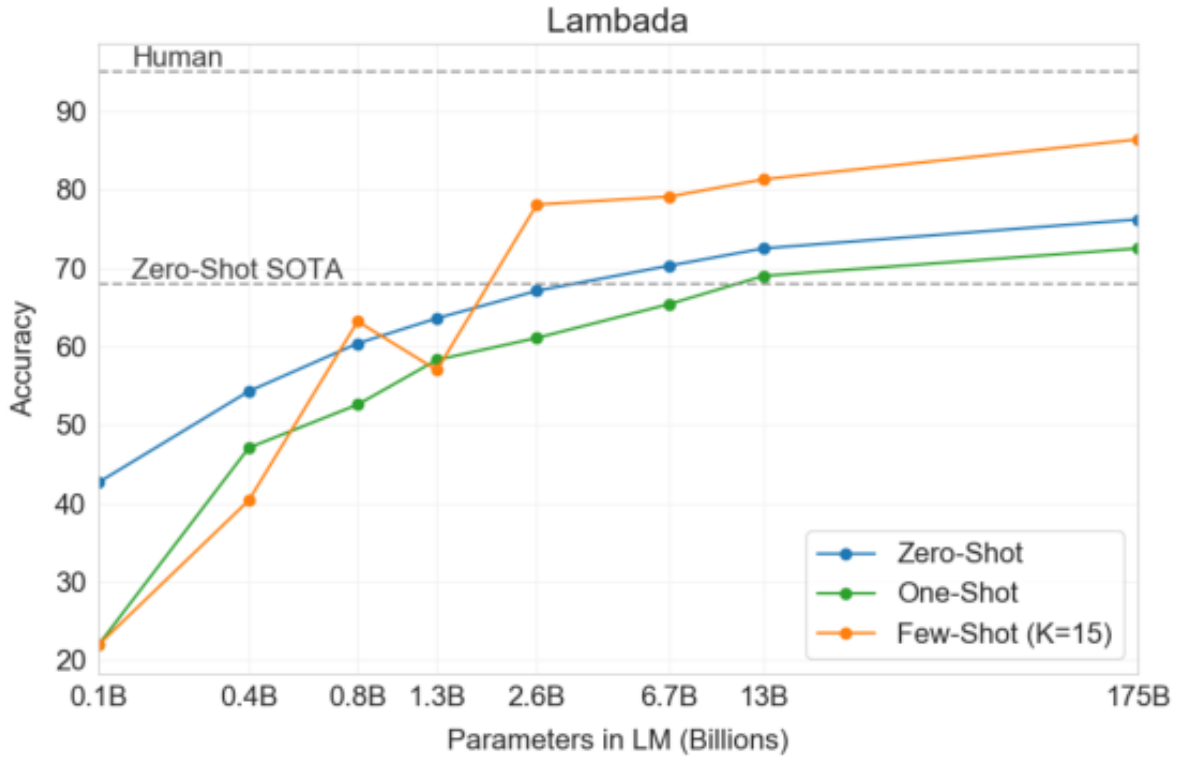


Few shot设置的三种预测方法

效果展示

效果展示

- 算法执行结果
  - 语言建模任务

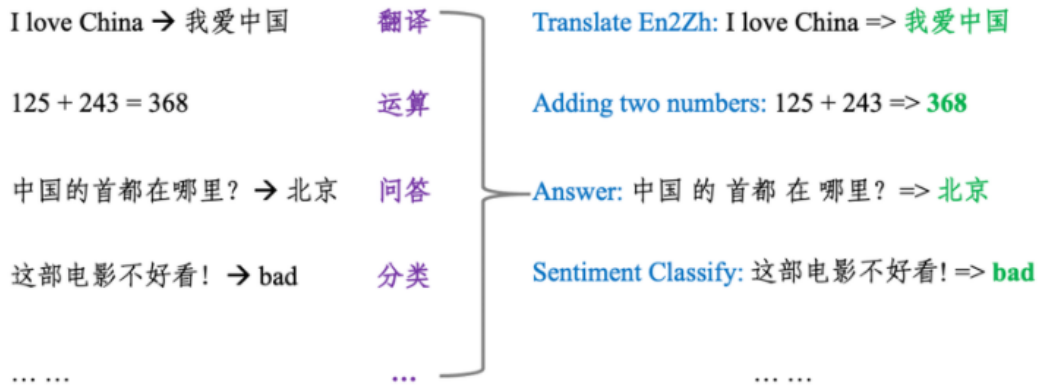


88%

在Lambada数据集语言建模的准确性对比

优劣分析

- 优点
  - 无需fine-tuning
  - 稀疏注意机制缓解了大规模模型训练代价
  - 网络结构庞大提高性能
  - 所有NLP任务转换为语言模型任务
- 缺点
  - 单向建模，建模能力受限
  - 规模过大，计算能力要求高



所有问题转换为语言生成问题

应用总结

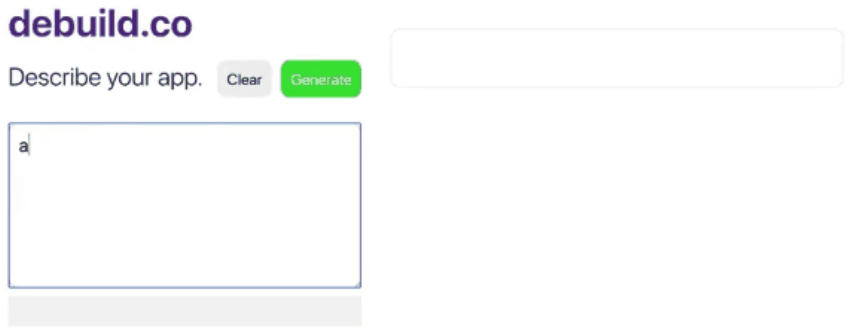
- 应用领域
  - 代码生成
  - 文本生成
  - 机器翻译
- 未来的发展
  - GPT3+微调
  - 拥有组合性推理的能力

```
Q: Which is heavier, a toaster or a pencil?
A: A pencil is heavier than a toaster.
```

问：铅笔和烤面包机哪一个更重

答：铅笔比烤面包机重

GPT3对基本逻辑问题的回答



代码自动生成网站debuild

Input: *The Night King slowly walked up to the three-eyed Raven. Jon was held up by the Ice Dragon and Daenerys was surrounded by Wights. The Night King reaches for his ice sword to kill Bran, but suddenly*

Output: *Hodor stops him, and he's holding the direwolf still with his big hands. "HODOR!" the raven cried. "HODOR! HODOR!" The black lord wore a lost look on his face. He thrust his stump through Bran's back and began fighting, rolling, spinning. Wights, giants, halflings, straw men. Bran fought with Summer and the Eagle and Blockhead Dog and the Stranger, but the Lord of Light would never shield a wight against a sword. The raven was saying that just climbing up Hodor's thick snow-white legs.*

GPT3改写的权游结局

参考资料

- Bengio, Yoshua, Ducharme, et al. A Neural Probabilistic Language Model.[J]. Journal of Machine Learning Research, 2003.
- Brown, Tom B., et al. "Language models are few-shot learners." arXiv preprint arXiv:2005.14165 (2020).
- https://www.bilibili.com/video/BV1a5411j7Sw?from=search&seid=11518669909063388668
- https://blog.csdn.net/iFlyAI/article/details/108867881
- https://blog.csdn.net/qq_27717921/article/details/99670843

谢谢！

大成若缺，其用不弊。大盈若冲，其用不穷。大直若屈。大巧若拙。大辩若讷。静胜躁，寒胜热。清静为天下正。